

XUIMS

The X-Window User Interface Management System at CERN

M.Vanden Eynden (E-mail : veym@dxcern.cern.ch)
CERN SL Controls Group
European Laboratory for Particle Physics
CH 1211 Geneva 23, Switzerland

ABSTRACT

The CERN X-Window User Interface Management System (XUIMS) is a modular and highly configurable software development environment allowing the interactive design, prototyping, and production of OSF/Motif Human Computer Interfaces (HCI). Fully compliant with the X11R5 and OSF/Motif industry standards, XUIMS covers complex software areas like the development of schematics, the visualization and on-line interactions with 2D and 3D scientific data, the display of relational database data and the direct access to CERN SPS and LEP accelerator equipment.

The guarantee of consistency across the applications and the encapsulation of complex functionality in re-usable and user-friendly components has also been implemented through the development of home made graphical objects (widgets) and templates.

The XUIMS environment is built with commercial software products integrated in the CERN SPS and LEP controls infrastructure with a very limited in-house effort.

Productivity and quality have been improved through less coding and better HCI prototyping.

INTRODUCTION

The X-Window and OSF/Motif industry standards have been used since 1992 to produce the Human Computer Interfaces (HCI) required for the control of the two CERN largest particle accelerators : SPS and LEP. The decision to base the HCI software activity on industry standards was made to avoid, as much as possible, vendor-specific developments and to guarantee thereby the software investment for a long time [1].

The desire to optimize the production of HCIs led to several market investigations and product evaluations. In 1994, a commercial User Interface Management System was selected to become the kernel of the CERN X-Windows User Interface Management System (XUIMS) [2].

X-WINDOWS AND OSF/MOTIF

The X-Windows and OSF/Motif software market has rapidly evolved during last two years and today offers a large choice of HCI software development environments. One could still consider the development of HCIs using only the standard X-Window and OSF/Motif libraires, but if a higher degree of abstraction and reusability is required, or if special features (i.e. 2D and 3D scientific data visualization) are needed, then using a powerful software development environment becomes mandatory. The next four sections summarise the different strategies that can be considered for the development of OSF/Motif interfaces.

X-WINDOWS AND OSF/MOTIF LIBRARIES

The X-Windows and OSF/Motif libraries (Xlib, XtIntrinsics, and Xm) are intended to be called from a C application and contain low level functions for the creation and manipulation of HCI components (i.e. buttons, labels and lists of elements). Any modification in the HCI requires a change in the C program (i.e. editing, compilation, and debugging development cycle). The final HCI C code is not understandable by any Interface Development Tool (IDT) or User Interface Management System (UIMS) and has therefore to be maintained manually.

OSF/MOTIF USER INTERFACE LANGUAGE

The OSF/Motif UIL language is a specification language for describing the initial state of a user interface for a Motif application. The specification describes the objects (i.e. menus, form boxes and buttons) used in the interface and specifies the functions to be called when the interface changes state as a result of user interaction. The main advantage of UIL is that it is understood by most IDTs and UIMSs. Nevertheless, UIL does not address the dynamic aspects of the interface (i.e. printing an information message when the user presses a button) and still requires the recompilation of the UIL file after each modification.

INTERFACE DEVELOPMENT TOOLS

Standard Interface Development Tools (IDT) help the developer to design the layout of the interface through a *What You See Is What You Get* (WYSIWYG) interface and to generate the corresponding C code. The application code has then to be inserted into stubs. The main disadvantage of the IDT is that the HCI behavior (i.e. putting an alarm message on the screen) must still be handled by the application code, leading to non-reusable software (see chapter about HCI development principles).

USER INTERFACE MANAGEMENT SYSTEMS

User Interface Management Systems (UIMS) add to the IDT functionality the possibility to design and prototype the dynamic behavior of the HCI. These tools can usually import UIL files and export UIL, C, or C++ HCI code. Their ability to address both the static and dynamic aspects of the HCI offers for the first time the possibility to put in place a real HCI software development lifecycle policy.

These considerations led us to buy a commercial UIMS and to enrich it with :

- new HCI components such as 2D and 3D graphics widgets,
- additional high level C functions (i.e. routines for direct SPS and LEP equipment access),
- templates for a common style and behaviour within the SPS and LEP operational environment.

THE XUIMS ARCHITECTURE

As shown in Fig 1, XUIMS is a highly modular and configurable environment composed of the following building blocks :

- *The XFaceMaker™ UIMS*. XFaceMaker[3] has a WYSIWYG editor for the interactive creation of HCI layouts.

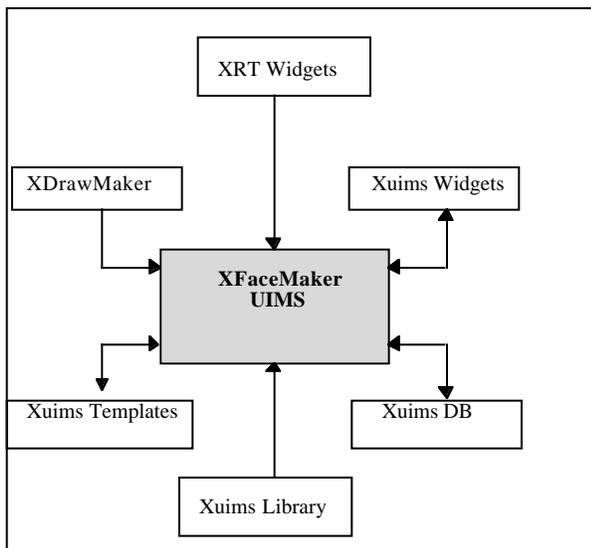


Fig 1 : The XUIMS Architecture

The HCI behavior is specified in the form of procedural scripts in a high-level, C-like programming language called FACE. The behavior can be tested within the tool using the built-in interpreter and debugger for the FACE scripting language. C, C++, and UIL code are generated by the tool. Additional C libraries can be linked into the UIMS kernel (i.e. libraries for SPS and LEP accelerator equipment access) in order to extend the basic functionalities of XFaceMaker. Working HCIs produced with XFaceMaker can run on any workstation without run-time fees (this is also the case for the other commercial products used in XUIMS). XFaceMaker also allows the programmer to define templates (composite objects) and to add new widgets to the standard OSF/Motif widgets set.

- *The XDrawMaker™ drawing package.* XDrawMaker [4] is a drawing editor enhanced by the ability to animate objects. When used with XFaceMaker, every object becomes a widget that can be included and manipulated in the HCI. Attributes of objects in complex schematics (see Fig 2) can be handled as any standard OSF/Motif widget. Almost any imaginable task can be handled with XDrawMaker.

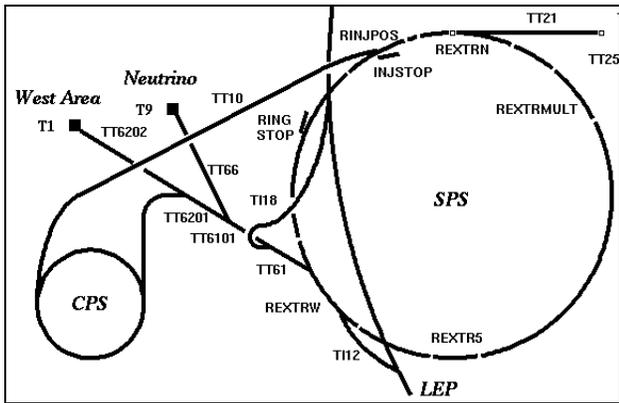


Fig 2 : Schematics of CERN's intricate particle beam network built with XDrawMaker.
Courtesy of the CERN SL Transfer Line project team.

- *The XRT™ 2D, 3D and Table widgets.* The XRT/2D widget can display virtually any type of two-dimensional graph, including barcharts, X-Y plots, pie charts, area graphs and logarithmic scientific charts. The XRT/3D widget can contour and zone 3D data. Users can dynamically rotate, zoom and scale views. The XRT/table widget allows developers to display and manipulate tabular data. The XRT widgets were selected for their excellent real-time performance. Fig 3 shows Xplore, the CERN off-line scientific data visualization package built with XUIMS and the XRT/2D widget.
- *The XUIMS library.* This C library contains additional functions used for direct access to the SPS and LEP accelerators equipment, for direct access to the UNIX file system, for providing on-line hypertext help on the WWW and for the handling of HCI components (i.e. flashing widgets, information messages, and cursor manipulations).
- *The XUIMS widgets.* Extending the OSF/Motif widgets set with XFaceMaker is a trivial exercise. New widgets and their associated code can be developed interactively with the WYSIWYG editor and added to the standard palette afterwards. This important feature has been used to create additional widgets for formatted numeric entries, for file printing and screen dumping and for providing help.
- *The XUIMS templates.* As shown in Fig 3, a template is a model, defined by a pre-designed object, that can be applied to other objects, called the instances of the template, which inherit the attributes defined in the template. A template can also be a composite object (i.e. a main window with predefined buttons and actions). Templates are very useful to encapsulate high-level functionality in a reusable way and to enforce consistency across the applications.

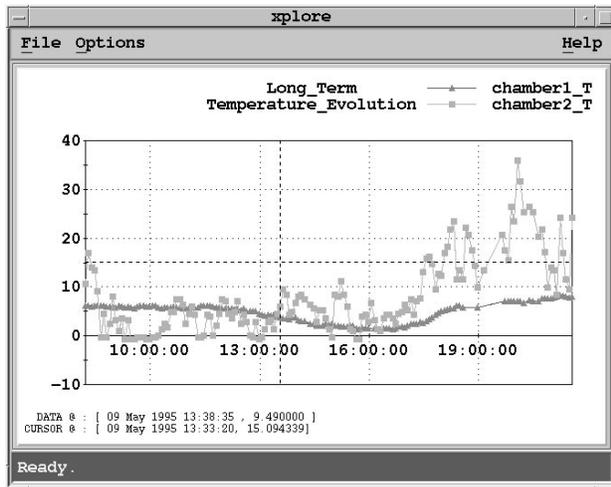


Fig 3 : The CERN Xplore off-line scientific data visualization Package built with the XUIMS 2D template.

- *The XUIMS resources database.* The main purpose of this database is to guarantee a standard behavior for all X-Window applications running in the SPS and LEP control room. Consistency is important both between applications and within a single application. Consistency helps the user to transfer familiar skills to new situations. The user can apply the knowledge learned from one application to another application, reducing the amount of learning and subsequent calls (i.e. an operator will easily recognize a push-button in an application if all pushbuttons have the same background and foreground colors). A primary XUIMS resources database [5] contains low priority definitions of resource values (i.e. fonts and colors) affecting all X-Window HCIs. If needed, a secondary resource database containing higher priority resource values can be provided by the developer for one given HCI.

THE HCI DESIGN WITH XUIMS

THE INTERFACE AND THE APPLICATION

The fundamental principle of HCI software development is the separation between the application and the interface [6]. The application can be seen as the set of routines and algorithms specific to the process or equipment to be controlled (i.e. routine for switching a vacuum pump ON and OFF). The interface can be seen as a set of windows through which the user will interact with the application.

In terms of reusability, one application code may be invoked by several interfaces based on different technologies (i.e. a C library used to control an equipment could be invoked either from an OSF/Motif operator interface, or from a C program running on a VT100 terminal).

A lack of separation between the application and interface software layers leads, in most cases to non-reusable software (i.e. equipment access code containing lowlevel OSF/Motif library calls). A functional specification of the data flows between the interface and the application layers is a mandatory step towards better software reusability and a better development lifecycle. The HCI development lifecycle described in the next section is a first step in this direction.

THE HCI DEVELOPMENT LIFECYCLE

As shown in Figure 4, a typical HCI development lifecycle with XUIMS [6][7] can be decomposed as follows :

- I. *The application concept.* This phase, also called the user requirements phase, is aimed at gathering quantifiable HCI requirements, identifying the future users of the system and drawing the HCI metaphor.
- II. *The functional specification.* This phase consists of the translation of an application concept (developed by a user) into a functional software specification. It should be independent of any particular windowing or graphical system. The

REFERENCES

- [1] The CERN PS and SL Controls Consolidation Project
CERN PS and SL Controls Groups
CERN Note PS/91-09 or SL/91-12

- [2] The CERN SL and ST User Interface Management System - XUIMS
Software User Manual.
P.Ninin, M.Vanden Eynden
CERN SL Note 94-53 (CO)

- [3] XFaceMaker User's Guide V3.0.
Non Standard Logics

- [4] XDrawMaker - The Animated Drawing Design Tool for the X Window System.
Non Standard Logics

- [5] The SL X-Window Migration Project
Resources Management Policy.
M.Tarrant, M.Vanden Eynden
CERN SL Note 93-77 (CO)

- [6] Software Engineering The PSS-05 Way.
Quality Systems & Software

- [7] Software Engineering Standards.
C.Mazza, J.Fairclough, B.Melton, D. De Pablo, A.Scheffer, R.Stevens - Prentice Hall

[™] XFaceMaker is a trademark of Non Standard Logics, France.

[™] XDrawMaker is a trademark of Non Standard Logics, France.

[™] XRT/2D, XRT/3D, and XRT/Table are trademarks of KL Group, Inc.